IN THE CLAIMS:

The following listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Original) A method for rendering graphics data, the method comprising:

receiving a scene graph, wherein the scene graph comprises a hierarchical group of objects;

generating a plurality of data structures, wherein each data structure corresponds to a particular type of object in the scene graph;

creating at least one thread that operates on each data structure, wherein at least a subset of the threads are configured to generate messages to communicate state changes to one or more of the data structures; and

generating multiple renderer threads that each render to a single canvas.

2. (Original) The method of claim 1, further comprising guaranteeing that opaque objects render before transparent objects.

3. (Previously Presented) The method of claim 1, further comprising granting each renderer thread a separate context.

4. (Original) A method for rendering graphics data, the method comprising:

receiving a scene graph, wherein the scene graph comprises a hierarchical group of objects;

generating a plurality of data structures, wherein each data structure corresponds to a particular type of object in the scene graph;

creating at least one thread that operates on each data structure, wherein at least a subset of the threads are configured to generate messages to communicate state changes to one or more of the data structures; and

generating multiple renderer threads, wherein each renderer thread renders to a different canvas.

5. (Original) A method for rendering graphics data, the method comprising:

receiving a scene graph, wherein the scene graph comprises a hierarchical group of objects;

generating a plurality of data structures, wherein each data structure corresponds to a particular type of object in the scene graph;

creating at least one thread that operates on each data structure, wherein at least a subset of the threads are configured to generate messages to communicate state changes to one or more of the data structures; and

generating multiple renderer threads, wherein each renderer resides on a separate graphics device.

6. (Original) The method of claim 5, wherein each renderer thread executed in parallel.

7. (Original) The method of claim 5, wherein each renderer thread is configured to issue swap buffer commands on all canvases associated with a single view at the same time.

8. (Original) The method of claim 5, wherein each renderer thread is configured to issue swap buffer commands on all canvases associated with a single view at substantially the same time.

9. (Original) The method of claim 5, wherein each renderer thread is configured to issue swap buffer commands on all canvases associated within as close to each other as possible.

10. (Previously Presented) A computer-readable medium comprising a computer program, wherein the computer program is executable by a computer to implement:

receiving a scene graph, wherein the scene graph comprises a hierarchical group of objects;

3

generating a plurality of data structures, wherein each data structure corresponds to a particular type of object in the scene graph;

creating at least one thread that operates on each data structure, wherein at least a subset of the threads are configured to generate messages to communicate state changes to one or more of the data structures; and

generating multiple renderer threads that each render to a single canvas.

11. (Previously Presented) The computer-readable medium of claim 10, wherein the computer program is an application program interface (API).

12. (Previously Presented) The computer-readable medium of claim 10, wherein the computer program is an application program interface (API) for the Java programming language.

13. (Previously Presented) A computer-readable medium comprising a computer program, wherein the computer program is executable by a computer to:

receive a scene graph comprising a plurality of graphical objects;

traverse the scene graph;

generate a plurality of data structures corresponding to the scene graph, wherein at least one of the data structures is a render bin;

determine which of a plurality of graphical objects are to be rendered;

create entries in the render bin that correspond to graphical objects that are to be rendered; and

submit the graphical objects listed in the render bin for rendering.

14. (Previously Presented) The computer-readable medium of claim 13, further comprising rendering the graphical objects listed in the render bin.

15. (Previously Presented) The computer-readable medium of claim 13, further comprising generating a rendering thread, wherein the rendering thread is configured to render the contents of the render bin.

16. (Previously Presented) The computer-readable medium of claim 13, wherein the entries in the render bin are pointers to memory corresponding to the graphical objects that are to be rendered.

17. (Previously Presented) The computer-readable medium of claim 13, wherein each data structure has a message queue configured to receive and store messages including data for updating the data structure.

18. (Previously Presented) The computer-readable medium of claim 14, wherein the scene graph is traversed only once, and wherein said rendering is performed more than once.

19. (Previously Presented) The computer-readable medium of claim 13, wherein additions to the scene graph generate updates to one or more of the data structures.

20. (Previously Presented) The computer-readable medium of claim 13, wherein the render bin has one or more render threads associated with it, wherein each render thread is configured to render a portion of the render bin.

21. (Previously Presented) The computer-readable medium of claim 13, wherein the render bin comprises an array of pointers pointing to the graphical objects that are to be rendered.

22. (Previously Presented) The computer-readable medium of claim 13, wherein the render bin comprises an array of graphical objects that are to be rendered.

23. (Previously Presented) The computer-readable medium of claim 13, wherein the render bin comprises a list of pointers pointing to the graphical objects that are to be rendered.

24. (Currently Amended) A <u>computer-implemented</u> method ~~for managing data that is to~~ ~~be rendered~~, the method comprising:

receiving a scene graph comprising a plurality of graphical objects;

generating a plurality of data structures corresponding to the scene graph, wherein at least one of the data structures is a render bin;

determining which of the graphical objects are to be rendered;

creating entries in the render bin that correspond to graphical objects that are to be rendered; and

submitting the graphical objects listed in the render bin for rendering.

25. (Currently Amended) The <u>computer-implemented</u> method of claim 24, wherein the data structures include: a geometry data structure, a transform data structure, and rendering environment data structure.

26. (Currently Amended) The <u>computer-implemented</u> method of claim 24, further comprising rendering the graphical objects having entries listed in the render bin.

27. (Currently Amended) The <u>computer-implemented</u> method of claim 24, further comprising generating a rendering thread, wherein the rendering thread is configured to render the graphical objects having entries in the render bin.

28. (Currently Amended) The <u>computer-implemented</u> method of claim 24, wherein the entries in the render bin are pointers to memory locations storing the graphical objects that are to be rendered.

29. (Currently Amended) The <u>computer-implemented</u> method of claim 24, wherein each data structure has a message queue configured to receive and store messages, wherein the messages include data for updating the data structure.

30. (Currently Amended) The computer-implemented method of claim 24, wherein the scene graph is traversed only once, and wherein said rendering is performed more than once.

31. (Currently Amended) The computer-implemented method of claim 24, wherein additions to the scene graph generate updates to one or more of the data structures.

32. (Currently Amended) The computer-implemented method of claim 24, wherein the render bin has one or more render threads associated with it, wherein each render thread is configured to render the graphical objects having entries in the render bin.

33. (Currently Amended) The computer-implemented method of claim 24, wherein the render bin comprises an array or list of pointers pointing to the graphical objects that are to be rendered.

34. (Currently Amended) The computer-implemented method of claim 24, wherein the render bin comprises an array of the graphical objects that are to be rendered.

35. (Original) A method for rendering graphics data, the method comprising:
   receiving data corresponding to objects in a virtual world;
   creating entries in a hierarchical scene graph corresponding to the objects;
   creating a plurality of data structures, wherein each data structure corresponds to a particular type of object in the virtual world;
   creating entries in the data structures corresponding to the objects;
   creating at least one thread for each data structure; and
   rendering the objects having entries in one of the data structures.

36. (Original) The method of claim 35, wherein the threads are configured to execute in parallel.

37. (Original) The method of claim 35, wherein at least one of the data structures is a render bin, and wherein one of the threads is a rendering thread.

38. (Original) The method of claim 37, wherein the scene graph comprises one or more views, wherein there is one render bin for each view.

39. (Original) The method of claim 37, wherein each data structure that is not a render bin has at least one working thread, wherein each working thread is configured to update the corresponding data structure.

40. (Original) The method of claim 35, wherein at least one of the data structures is a rendering environment data structure.

41. (Original) The method of claim 35, wherein at least one of the data structures is a transform data structure.

42. (Original) The method of claim 35, wherein at least one of the data structures is a geometry data structure.

43. (Original) The method of claim 35, wherein at least one of the data structures is a behavior structure that stores entries relating to object behavior.